3GPP TSG-T (Terminals) Meeting #13
Beijing, China, 19 - 21 September, 2001

Tdoc TP-010203

Source: T3

Title: Change Requests SIM/USIM characteristics (TS 51.011 / 31.102)

Document for: Approval

This document contains change requests to TS 51.011 and TS 31.102 as agreed by T3.

T3 Doc	Spec	CR	Rel	Cat	Subject
T3-010583	51.011	003	Rel-4	F	EF(EXT1): Clarification of Length Indicator for Additional Data
T3-010584	31.102	096	Rel-4	F	EF(EXT1): Clarification of Length Indicator for Additional Data
T3-010587	31.102	098	Rel-4	A	General Corrections
T3-010588	31.102	099	R99	F	General Corrections

CR-Form-v3

CHANGE REQUEST

| ⌘ | **51.011** CR **003** | ⌘ rev **-** | ⌘ Current version: | **4.1.0** | ⌘ |

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM ☐ ME/UE **X** Radio Access Network ☐ Core Network ☐

Title: ⌘	EF$_{EXT1}$: Clarification of Length Indicator for Additional Data	
Source: ⌘	T3	
Work item code: ⌘		***Date:*** ⌘ 2001-09-05
Category: ⌘ **F**		***Release:*** ⌘ REL-4

Use one of the following categories:	*Use one of the following releases:*
F *(essential correction)*	*2 (GSM Phase 2)*
A *(corresponds to a correction in an earlier release)*	*R96 (Release 1996)*
B *(Addition of feature),*	*R97 (Release 1997)*
C *(Functional modification of feature)*	*R98 (Release 1998)*
D *(Editorial modification)*	*R99 (Release 1999)*
Detailed explanations of the above categories can	*REL-4 (Release 4)*
be found in 3GPP TR 21.900.	*REL-5 (Release 5)*

Reason for change: ⌘	The use of byte 2 in EF$_{EXT1}$ (i.e. first byte of Extension data) is ambiguous in case of additional data (which in this case represents a length indicator). In the existing specification the length indicator byte could be interpreted in different ways: a) length byte exists only in the first record of a chain, indicating the length of all remaining bytes of ADN/SSC b) length byte exists in each record, indicating the length of ADN/SSC within the same record only c) length byte exists in each record, indicating the length of ADN/SSC from the current position to the end After several discussions, solution b) seems to be the intended one.
Summary of change: ⌘	1. clarification given for the length byte in case of record-chaining. 2. example modified (now two records for additional data including length indicator)
Consequences if not approved: ⌘	MEs may misinterpret SIMs which have stored ADN/SSC longer than 40 digits.

Clauses affected: ⌘	10.5.10
Other specs affected: ⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments: ⌘	This CR consists of 3 pages (including this cover page)

10.5.10 EF_{EXT1} (Extension1)

This EF contains extension data of an ADN/SSC, an MSISDN, or an LND. Extension data is caused by:
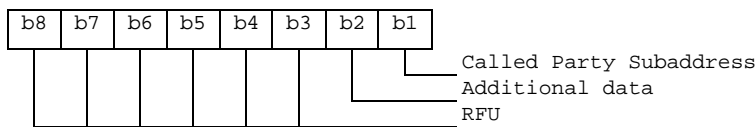
- an ADN/SSC (MSISDN, LND) which is greater than the 20 digit capacity of the ADN/SSC (MSISDN, LND) Elementary File or where common digits are required to follow an ADN/SSC string of less than 20 digits. The remainder is stored in this EF as a record, which is identified by a specified identification byte inside the ADN/SSC (MSISDN, LND) Elementary File. The EXT1 record in this case is specified as additional data;

- an associated called party subaddress. The EXT1 record in this case is specified as subaddress data.

Identifier: '6F4A'		Structure: linear fixed		Optional
Record length: 13 bytes		Update activity: low		
Access Conditions: 　READ　　　　　　CHV1 　UPDATE　　　　　CHV1 　INVALIDATE　　　ADM 　REHABILITATE　　ADM				
Bytes	Description		M/O	Length
1	Record type		M	1 byte
2 to 12	Extension data		M	11 bytes
13	Identifier		M	1 byte

- Record type

    Contents: type of the record

    Coding:

b8	b7	b6	b5	b4	b3	b2	b1

    Called Party Subaddress
    Additional data
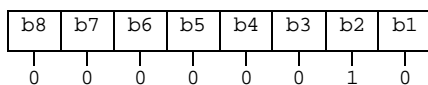   RFU

    b3-b8 are reserved and set to 0;
    a bit set to 1 identifies the type of record;
    only one type can be set;
    '00' indicates the type "unknown".

    The following example of coding means that the type of extension data is "additional data":

b8	b7	b6	b5	b4	b3	b2	b1
0	0	0	0	0	0	1	0

- Extension data

    Contents: Additional data or Called Party Subaddress depending on record type.

    Coding:

    Case 1, Extension1 record is additional data:

        The first byte of the extension data gives the number of bytes of the remainder of ADN/SSC (respectively MSISDN, LND). The coding of remaining bytes is BCD, according to the coding of ADN/SSC (MSISDN, LND). Unused nibbles at the end have to be set to 'F'. It is possible if the number of additional digits exceeds the capacity of the additional record to chain another record inside the EXT1 Elementary File by the identifier in byte 13. In this case byte 2 (first byte of the extension data) of all records for additional data within the same chain indicates the number of bytes ('01' to '0A') for ADN/SSC (respectively MSISDN, LND) within the same record unequal to 'FF'.

    Case 2, Extension1 record is Called Party Subaddress:

The subaddress data contains information as defined for this purpose in TS 04.08 [15]. All information defined in TS 04.08, except the information element identifier, shall be stored in the SIM. The length of this subaddress data can be up to 22 bytes. In those cases where two extension records are needed, these records are chained by the identifier field. The extension record containing the first part of the called party subaddress points to the record which contains the second part of the subaddress.

- Identifier

Contents: identifier of the next extension record to enable storage of information longer than 11 bytes.

Coding: record number of next record. 'FF' identifies the end of the chain.

EXAMPLE: Of a chain of extension records being associated to an ADN/SSC. The extension1 record identifier (Byte 14+X) of EF$_{ADN}$/SSC is set to 3.

No of Record	Type	Extension Data	Next	Record
:	:	:	:	
:	:	:	:	
Record 3	'02'	xx xx	'06'	►
Record 4	'xx'	xx xx	'xx'	
Record 5	'01'	xx xx	'FF'	◄
Record 6	'01'	xx xx	'05'	◄
:	:	:	:	
:	:	:	:	

EF$_{EXT1}$

Byte: 1	2	3	4	5	6	7	8	9	10	11	12	13
Record Type	Extension data											Identifier

Record 1	01	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	FF
Record 2	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
Record 3	02	0A	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	04
Record 4	02	04	xx	xx	xx	xx	FF	FF	FF	FF	FF	FF	06
Record 5	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
Record 6	01	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	01

In this example ADN/SSC is associated to additional data (records 3 and 4) which represents the last 27 or 28 digits of the whole ADN/SSC (the first 20 digits are stored in EF$_{ADN}$) and a called party subaddress whose length is more than 11 bytes (records 6 and 15).

CR-Form-v3

CHANGE REQUEST

| ⌘ | **31.102** CR **096** | ⌘ rev | **-** | ⌘ | Current version: | **4.1.0** | ⌘ |

For **HELP** *on using this form, see bottom of this page or look at the pop-up text over the* ⌘ *symbols.*

Proposed change affects: ⌘ (U)SIM ☐ ME/UE **X** Radio Access Network ☐ Core Network ☐

Title:	⌘	EF_EXT1: Clarification of Length Indicator for Additional Data

Source:	⌘	T3

Work item code:	⌘		*Date:* ⌘	2001-09-05

Category:	⌘	**F**	*Release:* ⌘	REL-4

	Use one of the following categories:	Use one of the following releases:
	F (essential correction)	2 (GSM Phase 2)
	A (corresponds to a correction in an earlier release)	R96 (Release 1996)
	B (Addition of feature),	R97 (Release 1997)
	C (Functional modification of feature)	R98 (Release 1998)
	D (Editorial modification)	R99 (Release 1999)
	Detailed explanations of the above categories can	REL-4 (Release 4)
	be found in 3GPP TR 21.900.	REL-5 (Release 5)

Reason for change: ⌘	The use of byte 2 in EF_EXT1 (i.e. first byte of Extension data) is ambiguous in case of additional data (which in this case represents a length indicator). In the existing specification the length indicator byte could be interpreted in different ways: a) length byte exists only in the first record of a chain, indicating the length of all remaining bytes of ADN/SSC b) length byte exists in each record, indicating the length of ADN/SSC within the same record only c) length byte exists in each record, indicating the length of ADN/SSC from the current position to the end After several discussions, solution b) seems to be the intended one.

Summary of change: ⌘	1. clarification given for the length byte in case of record-chaining. 2. example modified (now two records for additional data including length indicator)

Consequences if not approved: ⌘	MEs may misinterpret SIMs which have stored ADN/SSC longer than 40 digits.

Clauses affected:	⌘	4.4.2.4

Other specs affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications	

Other comments:	⌘	This CR consists of 3 pages (including this cover page)

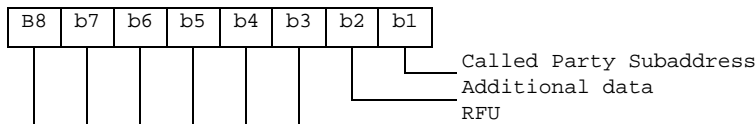## 4.4.2.4   EF$_{EXT1}$ (Extension1)

This EF contains extension data of an ADN/SSC. This EF shall always be present if the DF$_{Phonebook}$ is present.

Extension data is caused by:

-   an ADN/SSC which is greater than the 20 digit capacity of the ADN/SSC Elementary File or where common digits are required to follow an ADN/SSC string of less than 20 digits. The remainder is stored in this EF as a record, which is identified by a specified identification byte inside the ADN/SSC Elementary File. The EXT1 record in this case is specified as additional data;

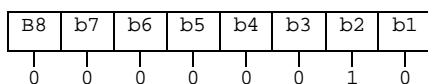-   an associated called party subaddress. The EXT1 record in this case is specified as subaddress data.

Identifier: '4FXX'	Structure: linear fixed		Optional
SFI: 'XX'			
Record length: 13 bytes		Update activity: low	

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM			

Bytes	Description	M/O	Length
1	Record type	M	1 byte
2 to 12	Extension data	M	11 bytes
13	Identifier	M	1 byte

-   Record type.
    Contents:
    -   type of the record.
    Coding:

B8	b7	b6	b5	b4	b3	b2	b1

    Called Party Subaddress
    Additional data
    RFU

-   b3-b8 are reserved and set to 0;

-   a bit set to 1 identifies the type of record;

-   only one type can be set;

-   '00' indicates the type "unknown".

The following example of coding means that the type of extension data is "additional data":

B8	b7	b6	b5	b4	b3	b2	b1
0	0	0	0	0	0	1	0

-   Extension data.
    Contents:
        additional data or Called Party Subaddress depending on record type.
    Coding:
        Case 1, Extension1 record is additional data:
        -   The first byte of the extension data gives the number of bytes of the remainder of ADN/SSC. The coding of remaining bytes is BCD, according to the coding of ADN/SSC. Unused nibbles at the end shall be set to 'F'. It is possible if the number of additional digits exceeds the capacity of the additional record to chain another record inside the EXT1 Elementary File by the identifier in byte 13. In this case byte 2 (first byte

of the extension data) of all records for additional data within the same chain indicates the number of bytes ('01' to '0A') for ADN/SSC (respectively MSISDN, LND) within the same record unequal to 'FF'.

Case 2, Extension1 record is Called Party Subaddress:

- The subaddress data contains information as defined for this purpose in 3G TS 24.008 [9]. All information defined in 3G TS 24.008, except the information element identifier, shall be stored in the USIM. The length of this subaddress data can be up to 22 bytes. In those cases where two extension records are needed, these records are chained by the identifier field. The extension record containing the first part of the called party subaddress points to the record which contains the second part of the subaddress.

- Identifier.

    Contents:

        identifier of the next extension record to enable storage of information longer than 11 bytes.

    Coding:

        record number of next record. 'FF' identifies the end of the chain.

- Example of a chain of extension records being associated to an ADN/SSC. The extension1 record identifier (Byte 14+X) of EF_ADN/SSC is set to 3.

No of Record	Type	Extension Data	Next	Record
:	:	:	:	
:	:	:	:	
Record 3	'02'	xx xx	'06'	►
Record 4	'xx'	xx xx	'xx'	
Record 5	'01'	xx xx	'FF'	◄
Record 6	'01'	xx xx	'05'	◄
:	:	:	:	
:	:	:	:	

EF_EXT1												
Byte: 1	2	3	4	5	6	7	8	9	10	11	12	13
Record Type	Extension data											Identifier
Record 1	01	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	FF
Record 2	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
Record 3	02	0A	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	04
Record 4	02	04	xx	xx	xx	xx	FF	FF	FF	FF	FF	FF	06
Record 5	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
Record 6	01	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	xx	01

In this example ADN/SSC is associated to additional data (records 3 and 4) which represents the last 27 or 28 digits of the whole ADN/SSC (the first 20 digits are stored in EF_ADN) and a called party subaddress whose length is more than 11 bytes (records 6 and 15).

3GPP T3 (USIM) Meeting #20
Marseille, France, 4-5 September, 2001

Tdoc T3-010587
Revision of T3-010563

CR-Form-v3

CHANGE REQUEST

⌘   **31.102 CR 098**   ⌘ rev **-** ⌘ Current version: **4.1.0**   ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘   (U)SIM **X**   ME/UE **X**   Radio Access Network ☐   Core Network ☐

Title: ⌘	General Corrections
Source: ⌘	T3
Work item code: ⌘	TEI ***Date:*** ⌘ 05/09/01
Category: ⌘ **F**	***Release:*** ⌘ REL-4

Use <u>one</u> of the following categories:
  F *(essential correction)*
  A *(corresponds to a correction in an earlier release)*
  B *(Addition of feature),*
  C *(Functional modification of feature)*
  D *(Editorial modification)*
Detailed explanations of the above categories can
be found in 3GPP TR 21.900.

Use <u>one</u> of the following releases:
  2   (GSM Phase 2)
  R96 (Release 1996)
  R97 (Release 1997)
  R98 (Release 1998)
  R99 (Release 1999)
  REL-4 (Release 4)
  REL-5 (Release 5)

Reason for change: ⌘	
Summary of change: ⌘	Replace 'XX' by 'YY' in the SFI of phonebook files. Section 6.4, replace Universal PIN key reference by '11' Replace "Access with PIN2 is limited to the USIM application" by " Access with PIN2 is limited to the ADF(USIM)" Indicate in section 6.4 that the usage qualifier value '00' is valid. IAP and GRP file update activity set to low Note on the search of the global phonebook Removal of restriction for the USIM to use global key reference '01' as PIN and '08' as PIN2. Removal of section 7.4 on VERIFY command restriction
Consequences if not approved: ⌘	Misinterpretation

Clauses affected: ⌘	4.4.2, 6.4, 7.4
Other specs Affected: ⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments: ⌘	

4.4.2　Contents of files at the DF PHONEBOOK level

The UICC may contain a global phonebook, or application specific phonebooks, or both in parallel. When both phonebook types co-exist, they are independent and no data is shared. In this case, it shall be possible for the user to select which phonebook the user would like to access.

It is recommended that the terminal searches for the global phonebook located under $DF_{TELECOM,}$ as its presence is not indicated anywhere in the USIM application.

The global phonebook is located in $DF_{PHONEBOOK}$ under $DF_{TELECOM.}$. Each specific USIM application phonebook is located in $DF_{PHONEBOOK}$ of its respective Application DF_{USIM}. The organisation of files in $DF_{PHONEBOOK}$ under DF_{USIM} and under $DF_{TELECOM}$ follows the same rules. Yet $DF_{PHONEBOOK}$ under DF_{USIM} may contain a different set of files than $DF_{PHONEBOOK}$ under $DF_{TELECOM}$. All phonebook related EFs are located under their respective $DF_{PHONEBOOK}$. USIM specific phonebooks are dedicated to application specific entries. Each application specific phonebook is protected by the application PIN.

If a GSM application resides on the UICC, the EFs ADN and EXT1 from one $DF_{PHONEBOOK}$ (defined at GSM application installation) are mapped to $DF_{TELECOM}$. Their file IDs are specified in GSM 11.11 [18], i.e. EF_{ADN} = '6F3A' and EF_{EXT1} = '6F4A', respectively. EF_{ADN} and EF_{PBR} shall always be present if the $DF_{Phonebook}$ is present. If any phonebook file other than EF_{ADN} or $EF_{EXT1,}$ is used, then EF_{PBC} shall be present.

If the UICC is inserted into a GSM terminal and a record in the phonebook has been updated, a flag in the entry control information in the EF_{PBC} is set from 0 to 1 by the card. If the UICC is later inserted into a 3G terminal again, the terminal shall check the flag in EF_{PBC} and if this flag is set, shall update the EF_{CC}, and then reset the flag. A set flag in EF_{PBC} results in a full synchronisation of the phonebook between an external entity and the UICC (if synchronisation is requested).

The EF structure related to the public phonebook is located under $DF_{PHONEBOOK}$ in $DF_{TELECOM}$. A USIM specific phonebook may exist for application specific entries. The application specific phonebook is protected by the application PIN. The organisation of files in the application specific phonebook follows the same rules as the one specified for the public phone book under $DF_{TELECOM}$. The application specific phonebook may contain a different set of files than the one in the public area under $DF_{TELECOM}$.

4.4.2.1　EF_{PBR} (Phone Book Reference file)

This file describes the structure of the phonebook. All EFs representing the phonebook are specified here, together with their file identifiers (FID) and their short file identifiers (SFI), if applicable.

Certain kinds of EFs can occur more than once in the phonebook, e.g. there may be two entities of Abbreviated Dialling Numbers, EF_{ADN} and EF_{ADN1}. For these kinds of EFs, no fixed FID values are specified. Instead, the value '4FXX' indicates that the value is to be assigned by the card issuer. These assigned values are then indicated in the associated TLV object in EF_{PBR}.

EFs stating an SFI value ('XXYY') in the description of their structure shall provide an SFI. The value shall be assigned by the card issuer and is indicated in the associated TLV object in EF_{PBR}.

The reference file is a file that contains information how the information in the different files is to be combined together to form a phone book entry. The reference file contains records. Each record specifies the structure of up to 254 entries in the phone book. Each phone book entry consists of data stored in files indicated in the reference file record. The entry structure shall be the same over all the records in the EF_{PBR}. If more than 254 entries are to be stored, a second record is needed in the reference file. The structure of a phone book entry is defined by different TLV objects that are stored in a reference file record. The reference file record structure describes the way a record in a file that is part of the phonebook is used to create a complete entry. Three different types of file linking exist.

- Type 1 files: Files that contain as many records as the reference/master file (EF_{ADN}, EF_{ADN1}) and are linked on record number bases (Rec1 -> Rec1). The master file record number is the reference.

- Type 2 files: Files that contain less entries than the master file and are linked via pointers in the index administration file (EF_{IAP}).

- Type 3 files are files that are linked by a record identifier within a record.

Table 4.1: Phone Book Reference file Constructed Tags

Tag Value	Constructed TAG Description
'A8'	Indicating files where the amount of records equal to master EF, type 1
'A9'	Indicating files that are linked using the index administration file, type 2. Order of pointer appearance in index administration EF is the same as the order of file IDs following this tag
'AA'	Indicating files that are linked using a record identifier, type 3. (The file pointed to is defined by the TLV object.)

The first file ID indicated using constructed Tag 'A8' is called the master EF. Access conditions for all other files in the index structure is set to the same as for the master EF unless otherwise specified.

File IDs indicated using constructed Tag 'A8' is a type 1 file and contains the same number of records as the first file that is indicated in the data part of this TLV object. All files following this Tag are mapped one to one using the record numbers/IDs of the first file indicated in this TLV object.

File IDs indicated using constructed Tag 'A9' are mapped to the master EF (the file ID indicated as the first data object in the TLV object using Tag 'A8') using the pointers in the index administration file. The order of the pointers in the index administration file is the same as the order of the file IDs presented after Tag 'A9'. If this Tag is not present in the reference file record the index administration file is not present in the structure. In case the index administration file is not present in the structure it is not indicated in the data following tag 'A8'.

File IDs indicated using constructed Tag 'AA' indicate files that are part of the reference structure but they are addressed using record identifiers within a record in one or more of the files that are part of the reference structure. The length of the tag indicates whether the file to be addressed resides in the same directory or if a path to the file is provided in the TLV object.

Type 2 and type 3 files contain records that may be shared between several phonebook entries (except when otherwise indicated). The terminal shall ensure that a shared record is emptied when the last phonebook entry referencing it is modified in such a way that it doesn't reference the record anymore.

NOTE: in the current version of the specification, only type 3 files contain records that may be shared.

Each constructed Tag contains a list of primitive Tags indicating the order and the kind of data (e.g. ADN, IAP,…) of the reference structure.

The primitive tag identifies clearly the type of data, its value field indicates the file identifier and, if applicable, the SFI value of the specified EF. That is, the length value of a primitive tag indicates if an SFI value is available for the EF or not:

- Length = '02' Value: 'FID (2 bytes)'

- Length = '03' Value: 'FID (2 bytes)', 'SFI (1 byte)'

Table 4.2: Tag definitions for the phone book kind of file

Tag Value	TAG Description
'C0'	EF_{ADN} data object
'C1'	EF_{IAP} data object
'C2'	EF_{EXT1} data object
'C3'	EF_{SNE} data object
'C4'	EF_{ANR} data object
'C5'	EF_{PBC} data object
'C6'	EF_{GRP} data object
'C7'	EF_{AAS} data object
'C8'	EF_{GAS} data object
'C9'	EF_{UID} data object
'CA'	EF_{EMAIL} data object
'CB'	EF_{CCP1} data object

Table 4.3 (below) lists the allowed types for each kind of file:

Table 4.3: Presence of files as type

File name	Type 1	Type 2	Type 3
EF_{AAS}			X
EF_{ADN}	X		
EF_{ANR}	X	X	
EF_{EMAIL}	X	X	
EF_{EXT1}			X
EF_{GAS}			X
EF_{GRP}	X		
EF_{IAP}	X		
EF_{PBC}	X		
EF_{SNE}	X	X	
EF_{UID}	X		
EF_{CCP1}			X

Phone Book Reference file EF$_{PBR}$ structure

Identifier: '4F30'		Structure: linear fixed		Conditional (see Note)
Record Length: X bytes		Update activity: low		
Access Conditions: READ PIN UPDATE ADM DEACTIVATE ADM ACTIVATE ADM				
Bytes	Description		M/O	Length
1 to X	TLV object(s) for indicating EFs that are part of the phone book structure		M	X bytes
NOTE: This file is mandatory if and only if DF$_{Phonebook}$ is present.				

At the end of each record, unused bytes, if any, shall be filled with 'FF'.

4.4.2.2 EF$_{IAP}$ (Index Administration Phone book)

This file is present if Tag 'A9' is indicated in the reference file.

The EF contains pointers to the different records in the files that are part of the phone book. The index administration file record number/ID is mapped one to one with the corresponding EF$_{ADN}$ (shall be record to record). The index administration file contains the same amount of records as EF$_{ADN}$. The order of the pointers in an EF$_{IAP}$ shall be the same as the order of file IDs that appear in the TLV object indicated by Tag 'A9' in the reference file record. The amount of bytes in a record is equal to the number of files indicated the EF$_{PBR}$ following tag 'A9'.

The value 'FF' is an invalid record number/ID and is used in any location in to indicate that no corresponding record in the indicated file is available.

The content of EF$_{IAP}$ is set to 'FF' at the personalisation stage.

Index administration file EF$_{IAP}$ structure

Identifier: '4FXX'	Structure: linear fixed	Conditional (see Note)
SFI: '~~XX~~YY'		
Record Length: X bytes	Update activity: low~~high~~	
Access Conditions: READ　　　　　　　PIN UPDATE　　　　　　PIN DEACTIVATE　　　　ADM ACTIVATE　　　　　ADM		

Bytes	Description	M/O	Length
1	Record number of the first object indicated after Tag 'A9'	M	1 byte
2	Record number of the second object indicated after Tag 'A9'	M	1 byte
X	Record number of the xth object indicated after Tag 'A9'	M	1 byte
NOTE:	This file is mandatory if and only if type 2 files are present.		

4.4.2.3　　EF$_{ADN}$ (Abbreviated dialling numbers)

This EF contains Abbreviated Dialling Numbers (ADN) and/or Supplementary Service Control strings (SSC). In addition it contains identifiers of associated network/bearer capabilities and identifiers of extension records. It may also contain an associated alpha-tagging.

Identifier: '4FXX'	Structure: linear fixed	Conditional (see Note)
SFI: '~~XX~~YY'		
Record length: X+14 bytes	Update activity: low	
Access Conditions: READ　　　　　　　PIN UPDATE　　　　　　PIN DEACTIVATE　　　　ADM ACTIVATE　　　　　ADM		

Bytes	Description	M/O	Length
1 to X	Alpha Identifier	O	X bytes
X+1	Length of BCD number/SSC contents	M	1 byte
X+2	TON and NPI	M	1 byte
X+3 to X+12	Dialling Number/SSC String	M	10 bytes
X+13	Capability/Configuration1 Identifier	M	1 byte
X+14	Extension1 Record Identifier	M	1 byte
NOTE:	This file is mandatory if and only if DF$_{PHONEBOOK}$ is present.		

- Alpha Identifier.
  Contents:
  - Alpha-tagging of the associated dialling number.
  Coding:
  - this alpha-tagging shall use
    either:
    - the SMS default 7-bit coded alphabet as defined in 3G TS 23.038 [5] with bit 8 set to 0. The alpha identifier shall be left justified. Unused bytes shall be set to 'FF'.
    or:
    - one of the UCS2 coded options as defined in the annex of 3G TS 31.101 [11].

NOTE 1: The value of X may be from zero to 241. Using the command GET RESPONSE the ME can determine the value of X.

- - Length of BCD number/SSC contents.
    Contents:
    - - this byte gives the number of bytes of the following two data items containing actual BCD number/SSC information. This means that the maximum value is 11, even when the actual ADN/SSC information length is greater than 11. When an ADN/SSC has extension, it is indicated by the extension1 identifier being unequal to 'FF'. The remainder is stored in the EF_{EXT1} with the remaining length of the additional data being coded in the appropriate additional record itself (see subclause 4.4.2.4).
    Coding:
    - - according to 3G TS 24.008 [9].
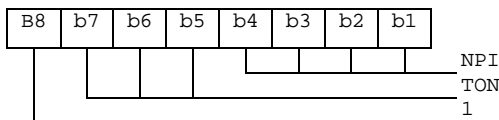
- - TON and NPI.
    Contents:
    - - Type of number (TON) and numbering plan identification (NPI).
    Coding:
    - - according to 3G TS 24.008 [9]. If the Dialling Number/SSC String does not contain a dialling number, e.g. a control string deactivating a service, the TON/NPI byte shall be set to 'FF' by the ME (see note 2).

- NOTE 2: If a dialling number is absent, no TON/NPI byte is transmitted over the radio interface (see 3G TS 24.008 [9]). Accordingly, the ME should not interpret the value 'FF' and not send it over the radio interface.

```
B8   b7   b6   b5   b4   b3   b2   b1
                     |    |    |    |____ NPI
      |    |    |_____ TON
      |_____ 1
```

- - Dialling Number/SSC String
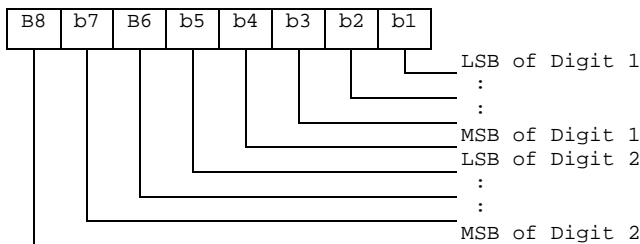   Contents:
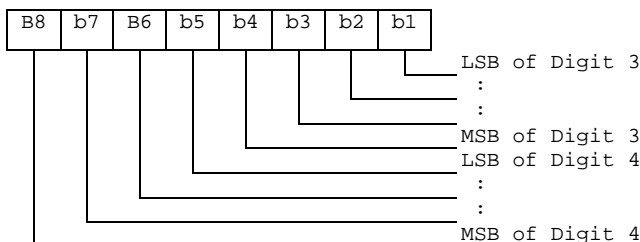   - - up to 20 digits of the telephone number and/or SSC information.
    Coding:
    - - according to 3G TS 24.008 [9] , 3G TS 22.030 [4] and the extended BCD-coding (see table 4.4). If the telephone number or SSC is longer than 20 digits, the first 20 digits are stored in this data item and the remainder is stored in an associated record in the EF_{EXT1}. The record is identified by the Extension1 Record Identifier. If ADN/SSC require less than 20 digits, excess nibbles at the end of the data item shall be set to 'F'. Where individual dialled numbers, in one or more records, of less than 20 digits share a common appended digit string the first digits are stored in this data item and the common digits stored in an associated record in the EF_{EXT1}. The record is identified by the Extension 1 Record Identifier. Excess nibbles at the end of the data item shall be set to 'F'.

Byte X+3

```
B8   b7   B6   b5   b4   b3   b2   b1
                           |    |    |____ LSB of Digit 1
                           |    |_____ :
                           |_____ :
                     |_____ MSB of Digit 1
                |_____ LSB of Digit 2
           |_____ :
      |_____ :
 |_____ MSB of Digit 2
```

Byte X+4:

```
B8   b7   B6   b5   b4   b3   b2   b1
                           |    |    |____ LSB of Digit 3
                           |    |_____ :
                           |_____ :
                     |_____ MSB of Digit 3
                |_____ LSB of Digit 4
           |_____ :
      |_____ :
 |_____ MSB of Digit 4
```

etc.

- Capability/Configuration1 Identifier.
  Contents:
  - capability/configuration identification byte. This byte identifies the number of a record in the EF$_{CCP1}$ containing associated capability/configuration parameters required for the call. The use of this byte is optional. If it is not used it shall be set to 'FF'.
  Coding:
  - binary.

- Extension1 Record Identifier.
  Contents:
  - extension1 record identification byte. This byte identifies the number of a record in the EF$_{EXT1}$ containing an associated called party subaddress or additional data. The use of this byte is optional. If it is not used it shall be set to 'FF'.
  - if the ADN/SSC requires both additional data and called party subaddress, this byte identifies the additional record. A chaining mechanism inside EF$_{EXT1}$ identifies the record of the appropriate called party subaddress (see subclause 4.4.2.4).
  Coding:
  - binary.

NOTE 3: EF$_{ADN}$ in the public phone book under DF$_{TELECOM}$ may be used by USIM, GSM and also other applications in a multi-application card. If the non-GSM application does not recognise the use of Type of Number (TON) and Number Plan Identification (NPI), then the information relating to the national dialling plan shall be held within the data item dialling number/SSC and the TON and NPI fields set to UNKNOWN. This format would be acceptable for 3G operation and also for the non-GSM application where the TON and NPI fields shall be ignored.

EXAMPLE: SIM storage of an International Number using E.164 [22] numbering plan.

	TON	NPI	Digit field.
USIM application	001	0001	abc...
Other application compatible with 3G	000	0000	xxx...abc...

where "abc..." denotes the subscriber number digits (including its country code), and "xxx..." denotes escape digits or a national prefix replacing TON and NPI.

NOTE 4: When the ME acts upon the EF$_{ADN}$ with a SEARCH RECORD command in order to identify a character string in the alpha-identifier, it is the responsibility of the ME to ensure that the number of characters used as SEARCH RECORD parameters are less than or equal to the value of X if the MMI allows the user to offer a greater number.

Table 4.4: Extended BCD coding

BCD Value	Character/Meaning
'0'	"0"
:	:
'9'	"9"
'A'	"*"
'B'	"#"
'C'	DTMF Control digit separator (GSM 02.07 [17]).
'D'	"Wild" value. This will cause the MMI to prompt the user for a single digit (see GSM 02.07 [17]).
'E'	RFU.
'F'	Endmark e.g. in case of an odd number of digits.

BCD values 'C', 'D' and 'E' are never sent across the radio interface.

NOTE 5: A second or subsequent 'C' BCD value will be interpreted as a 3 second PAUSE (see GSM 02.07 [17]).
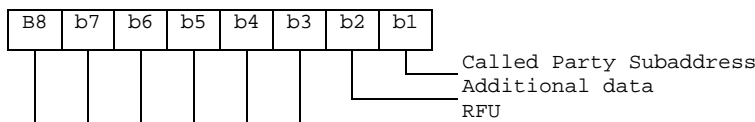
4.4.2.4   EF$_{EXT1}$ (Extension1)

This EF contains extension data of an ADN/SSC. . This EF shall always be present if the DF$_{Phonebook}$ is present.

Extension data is caused by:

- an ADN/SSC which is greater than the 20 digit capacity of the ADN/SSC Elementary File or where common digits are required to follow an ADN/SSC string of less than 20 digits. The remainder is stored in this EF as a record, which is identified by a specified identification byte inside the ADN/SSC Elementary File. The EXT1 record in this case is specified as additional data;

- an associated called party subaddress. The EXT1 record in this case is specified as subaddress data.
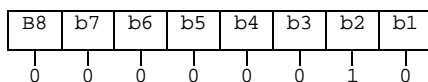
Identifier: '4FXX'		Structure: linear fixed		Optional
SFI: 'XXYY'				
Record length: 13 bytes		Update activity: low		

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM				

Bytes	**Description**		**M/O**	**Length**
1	Record type		M	1 byte
2 to 12	Extension data		M	11 bytes
13	Identifier		M	1 byte

- Record type.
  Contents:
  - type of the record.
  Coding:

- b3-b8 are reserved and set to 0;

- a bit set to 1 identifies the type of record;

- only one type can be set;

- '00' indicates the type "unknown".

The following example of coding means that the type of extension data is "additional data":

- Extension data.
  Contents:
    additional data or Called Party Subaddress depending on record type.
  Coding:
    Case 1, Extension1 record is additional data:
    - The first byte of the extension data gives the number of bytes of the remainder of ADN/SSC. The coding of remaining bytes is BCD, according to the coding of ADN/SSC. Unused nibbles at the end shall be set to 'F'. It is possible if the number of additional digits exceeds the capacity of the additional record to chain another record inside the EXT1 Elementary File by the identifier in byte 13.

Case 2, Extension1 record is Called Party Subaddress:
- The subaddress data contains information as defined for this purpose in 3G TS 24.008 [9]. All information defined in 3G TS 24.008, except the information element identifier, shall be stored in the USIM. The length of this subaddress data can be up to 22 bytes. In those cases where two extension records are needed, these records are chained by the identifier field. The extension record containing the first part of the called party subaddress points to the record which contains the second part of the subaddress.

- Identifier.
   Contents:
      identifier of the next extension record to enable storage of information longer than 11 bytes.
   Coding:
      record number of next record. 'FF' identifies the end of the chain.

- Example of a chain of extension records being associated to an ADN/SSC. The extension1 record identifier (Byte 14+X) of ADN/SSC is set to 3.

In this example ADN/SSC is associated to additional data (record 3) and a called party subaddress whose length is more than 11 bytes (records 6 and 5).

4.4.2.5 EF$_{PBC}$ (Phone Book Control)

This EF contains control information related to each entry in the phone book. This EF contains as many records as the EF$_{ADN}$ associated with it (shall be record to record). Each record in EF$_{PBC}$ points to a record in its EF$_{ADN}$. This file indicates the control information and the hidden information of each phone book entry.

The content of EF$_{PBC}$ is linked to the associated EF$_{ADN}$ record by means of the ADN record number/ID (there is a one to one mapping of record number/identifiers between EF$_{PBC}$ and EF$_{ADN}$).

Structure of control file EF$_{PBC}$

Identifier: '4FXX'	Structure: linear fixed		Conditional (see Note)
SFI: '~~XX~~YY'			
Record length: 2 bytes		Update activity: low	

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM			

Bytes	Description	M/O	Length
1	Entry Control Information	M	1 byte
2	Hidden Information	M	1 byte

NOTE: This file is mandatory if and only if one or both of the following is true:
               - hidden entries are supported
               - a GSM SIM application is supported in the UICC.			

- Entry Control Information.
   Contents:
   - provides some characteristics about the phone book entry (eg modification by a GSM mobile).

Coding:

b8	B7	b6	B5	b4	B3	b2	B1

Modified by GSM phone '1', no change '0'
RFU (see 3G TS 31.101)

- Hidden Information.
  Contents:
    indicates to which USIM application of the UICC this phone book entry belongs, so that the corresponding secret code can be verified to display the phone book entry. If the secret code is not verified, then the phone book entry is hidden.
  Coding:
    '00' – the phone book entry is not hidden;
    'xx' – the phone book entry is hidden. 'xx' is the record number in EF_{DIR} of the associated USIM application.

4.4.2.6 EF$_{GRP}$ (Grouping file)

This EF contains the grouping information for each phone book entry. This file contains as many records as the associated EF$_{ADN}$. Each record contains a list of group identifiers, where each identifier can reference a group to which the entry belongs.

Structure of grouping file EF$_{GRP}$

Identifier: '4FXX'		Structure: linear fixed		Conditional (see Note)
SFI: '~~XX~~YY'				
Record Length: X bytes (1 ≤ X ≤10)			Update activity: low~~high~~	

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM				

Bytes	Description		M/O	Length
1	Group Name Identifier 1		M	1 byte
2	Group Name Identifier 2		O	1 byte
X	Group Name Identifier X		O	1 byte
NOTE: This file is mandatory if and only if EF$_{GAS}$ is present.				

- Group Name Identifier x.

  Content:

  - indicates if the associated entry is part of a group, in that case it contains the record number of the group name in EF$_{GAS}$.

  - One entry can be assigned to a maximum of 10 groups.

  Coding:

  - '00' – no group indicated;
    'XX' – record number in EF$_{GAS}$ containing the alpha string naming the group of which the phone book entry is a member.

4.4.2.7 EF$_{AAS}$ (Additional number Alpha String)

This file contains the alpha strings that are associated with the user defined naming tags for additional numbers referenced in EF$_{ANR}$.

Structure of EF$_{AAS}$

Identifier: '4FXX'	Structure: linear fixed		Optional
SFI: -			
Record length: X bytes		Update activity: low	

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM			

Bytes	Description	M/O	Length
1 to X	Alpha text string	M	X bytes

- Alpha text string.

    Content:

    - user defined text for additional number.

    Coding:

    - same as the alpha identifier in EF$_{ADN}$.

4.4.2.8 EF$_{GAS}$ (Grouping information Alpha String)

This file contains the alpha strings that are associated with the group name referenced in EF$_{GRP}$.

Structure of EF$_{GAS}$

Identifier: '4FXX'	Structure: linear fixed		Conditional (see Note)
SFI: -			
Record length: X bytes		Update activity: low	

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM			

Bytes	Description	M/O	Length
1 to X	Alpha text string	M	X bytes
NOTE: This file is mandatory if and only if EF$_{GRP}$ is present.			

- Alpha text string

    Content:

    - group names.

    Coding:

    - same as the alpha identifier in EF$_{ADN}$.

4.4.2.9 EF$_{ANR}$ (Additional Number)

Several phone numbers and/or Supplementary Service Control strings (SSC) can be attached to one EF$_{ADN}$ record, using one or several EF$_{ANR}$. The amount of additional number entries may be less than or equal to the amount of records in EF$_{ADN}$. The EF structure is linear fixed. Each record contains an additional phone number or Supplementary Service Control strings (SSC). This record cannot be shared between several phonebook entries. The first byte indicates whether

the record is free or the type of additional number referring to the record number in EF$_{AAS}$, containing the text to be displayed. The following part indicates the additional number and the reference to the associated record in the EF$_{ADN}$ file. In addition it contains identifiers of associated network/bearer capabilities and identifiers of extension records.

Structure of EF$_{ANR}$

Identifier: '4FXX'		Structure: linear fixed		Optional	
SFI: '~~XX~~YY'					
Record length: 15 or 17 bytes			Update activity: low		
Access Conditions: 　　READ　　　　　　　PIN 　　UPDATE　　　　　 PIN 　　DEACTIVATE　　 ADM 　　ACTIVATE　　　　ADM					
Bytes	Description		M/O	Length	
1	Additional Number identifier		M	1 byte	
2	Length of BCD number/SSC contents		M	1 byte	
3	TON and NPI		M	1 byte	
4 to 13	Additional number/SSC String		M	10 bytes	
14	Capability/Configuration1 Identifier		M	1 byte	
15	Extension1 Record Identifier		M	1 byte	
16	ADN file SFI		C	1 byte	
17	ADN file Record Identifier		C	1 byte	
NOTE:　　The fields marked C above are mandatory if and only if the file is not type 1 (as specified in EF$_{PBR}$)					

- Additional Number Identifier

    Content:

    - describes the type of the additional number defined in the file EF$_{AAS}$.

    Coding:

    - '00' – no additional number description;
      'xx' – record number in EF$_{AAS}$ describing the type of number (e.g. "FAX");
      'FF' – free record.

- Length of BCD number/SSC contents

    Contents:

    - this byte gives the number of bytes of the following two data items containing actual BCD number/SSC information. This means that the maximum value is 11, even when the actual additional number/SSC information length is greater than 11. When the additional number/SSC has extension, it is indicated by the extension1 identifier being unequal to 'FF'. The remainder is stored in the EF$_{EXT1}$ with the remaining length of the additional data being coded in the appropriate additional record itself (see subclause 4.4.2.4).

    Coding:

    - same as the length of BCD number/SSC string byte in EF$_{ADN.}$

- TON and NPI.

    Contents:

    - Type of number (TON) and numbering plan identification (NPI).

    Coding:

    - same as the TON and NPI byte in EF$_{ADN.}$

- Additional number/SSC string

    Content:

    - up to 20 digits of the additional phone number and/or SSC information linked to the phone book entry.

    Coding:

    - same as the dialling number /SSC string in EF_{ADN}.

- Capability/Configuration1 Identifier.

    Contents:

    - This byte identifies the number of a record in the EF_{CCP1} containing associated capability/configuration parameters required for the call. The use of this byte is optional. If it is not used it shall be set to 'FF'.

    Coding:

    - binary.

- Extension1 Record Identifier.

    Contents:

    - extension1 record identification byte. This byte identifies the number of a record in the EF_{EXT1} containing an associated called party subaddress or additional data. The use of this byte is optional. If it is not used it shall be set to 'FF'.

        if the number requires both additional data and called party subaddress, this byte identifies the additional record. A chaining mechanism inside EF_{EXT1} identifies the record of the appropriate called party subaddress (see subclause 4.4.2.4).

    Coding:

    - binary.

- ADN file SFI.

    Content:

    - Short File identifier of the associated EF_{ADN} file.

    Coding:

    - as defined in the UICC specification.

- ADN file Record Identifier

    Content:

    - record identifier of the associated phone book entry.

    Coding:

    - 'xx' – record identifier of the corresponding ADN record.

4.4.2.10 EF$_{SNE}$ (Second Name Entry)

The phone book also contains the option of a second name entry. The amount of second name entries may be less than or equal to the amount of records in EF_{ADN}. Each record contains a second name entry. This record cannot be shared between several phonebook entries.

Structure of EF$_{SNE}$

Identifier: '4FXX'		Structure: linear fixed		Optional
SFI: '~~XX~~YY'				
Record length: X or X+2 bytes		Update activity: low		
Access Conditions: 　READ　　　　　　PIN 　UPDATE　　　　　PIN 　DEACTIVATE　　　ADM 　ACTIVATE　　　　ADM				
Bytes	Description		M/O	Length
1 to X	Alpha Identifier of Second Name		M	X bytes
X+1	ADN file SFI		C	1 byte
X+2	ADN file Record Identifier		C	1 byte
NOTE: The fields marked C above are mandatory if and only if the file is not type 1 (as specified in EF$_{PBR}$)				

- Alpha Identifier of Second Name.

  Content:

  - string defining the second name of the phone book entry.

  Coding:

  - as the alpha identifier for EF$_{ADN}$.

- ADN file SFI.

  Content:

  - Short File identifier of the associated EF$_{ADN}$ file.

  Coding:

  - as defined in the UICC specification.

- ADN file Record Identifier

  Content:

  record identifier of the associated phone book entry.

  Coding:

  'xx' – record identifier of the corresponding ADN record.

4.4.2.11　EF$_{CCP1}$ (Capability Configuration Parameters 1)

This EF contains parameters of required network and bearer capabilities and ME configurations associated with a call established using a phone book entry.

Structure of EF_CCP1

Identifier: '''4FXX'	Structure: linear fixed		Optional
SFI: '~~XX~~YY'			
Record length: X bytes, X ≥ 15		Update activity: low	
Access Conditions: 　READ　　　　　　　　PIN 　UPDATE　　　　　　　PIN 　DEACTIVATE　　　　ADM 　ACTIVATE　　　　　　ADM			
Bytes	Description	M/O	Length
1 to X	Bearer capability information element	M	X bytes

- Bearer capability information element.

    Contents and Coding:

    - see 3G TS 24.008 [9]. The Information Element Identity (IEI) shall be excluded; i.e. the first byte of the EF_CCP1 record shall be Length of the bearer capability contents.

"- unused bytes are filled with 'FF'

4.4.2.12 Phone Book Synchronisation

To support synchronisation of phone book data with other devices, the USIM may provide the following files to be used by the synchronisation method: a phone book synchronisation counter (PSC), a unique identifier (UID) and change counter (CC) to indicate recent changes.

If synchronisation is supported in the phonebook, then EF_PSC, EF_UID, EF_PUID and EF_CC are all mandatory.

4.4.2.12.1 EF_UID (Unique Identifier)

The EF_UID is used to uniquely identify a record and to be able to keep track of the entry in the phone book. The terminal assigns the (UID) when a new entry is created. The value of the UID does not change as long as the value of the PBID remains the same. The UID shall remain on the UICC, in EF_UID, until the PBID is regenerated. This means that when a phone book entry is deleted, the content of the linked information (eg ADN, E-MAIL,..) shall be set to the personalization value 'FF…FF'. But the UID-value of the deleted record shall not be used when a new entry is added to the phonebook until the PBID is regenerated, but it shall be set to a new value.

If/when the PBID is regenerated, all UIDs for the entry in the phone book shall be assigned new values starting from 1. The new value of the UID for each entry shall then be kept until the PBID is regenerated again.

Structure of EF_UID

Identifier: '4FXX'	Structure: linear fixed		Conditional (see Note)
SFI: '~~XX~~YY'			
Record length: 2 bytes		Update activity: low	
Access Conditions: 　READ　　　　　　　　PIN 　UPDATE　　　　　　　PIN 　DEACTIVATE　　　　ADM 　ACTIVATE　　　　　　ADM			
Bytes	Description	M/O	Length
1 to 2	Unique Identifier (UID) of Phone Book Entry	M	2 bytes
NOTE:　This file is mandatory if and only if synchronisation is supported in the phonebook.			

- Unique Identifier of Phone Book Entry.

    Content:

    - number to unambiguously identify the phone book entry for synchronisation purposes.

    Coding:

    - hexadecimal value. At initialisation all UIDs are personalised to "00 00" (i.e. empty).

4.4.2.12.2 EF$_{PSC}$ (Phone book Synchronisation Counter)

The phone book synchronisation counter (PSC) is used by the ME to construct the phone book identifier (PBID) and to determine whether the accessed phone book is the same as the previously accessed phone book or if it is a new unknown phone book (might be the case that there is one phonebook under DF-telecom and one phone book residing in a USIM-application). If the PSC is unknown, a full synchronisation of the phone book will follow.

The PSC is also used to regenerate the UIDs and reset the CC to prevent them from running out of range. When the UIDs or the CC has reached its maximum value, a new PSC is generated. This leads to a scenario where neither the CC nor the UIDs will run out of range.

The PSC shall be regenerated by the terminal if one of the following situation applies:
- the values of the UIDs have run out of range;
- the whole phone book has been reset/deleted;
- the value of the CC has run out of range.

Structure of EF$_{PSC}$

Identifier: '4F22'	Structure: transparent		Conditional (see Note)
SFI: '~~XX~~YY'			
File size: 4 bytes		Update activity: low	
Access Conditions: READ PIN UPDATE PIN DEACTIVATE ADM ACTIVATE ADM			

Bytes	Description	M/O	Length
1 to 4	Phone book synchronisation counter (PSC)	M	4 bytes
NOTE:	This file is mandatory if and only if synchronisation is supported in the phonebook.		

- PSC: Unique synchronisation counter of Phone Book.

    Content:

    number to unambiguously identify the status of the phone book for synchronisation purposes.

    Coding:

    hexadecimal value.

    The phone book identifier (PBID) coding based on the EF$_{PSC}$ is described hereafter:

- For a phone book residing in DF-telecom:

    - PBID = ICCid (10bytes) "fixed part" + 4 bytes (in EF$_{PSC}$) "variable part".

- For a phone book residing in an USIM application:

    - PBID = 10 last bytes of (ICCid XOR AID) "fixed part" + 4 bytes (in EF$_{PSC}$) "variable part".

To be able to detect if the PSC needs to be regenerated (i.e. the variable part) the following test shall be made by the terminal before for each update of either the CC or the assignment of a new UID:

- Each time the terminal has to increment the value of the UID the following test is needed:

    - If UID = 'FF FF' then.

        {Increment **PSC** mod 'FF FF FF FF'; all the UIDs shall be regenerated}.

- Each time the terminal has to increment the value of CC the following test is needed:

    If CC = 'FF FF' then.

    {Increment **PSC** mod 'FF FF FF FF' ; CC=0001}.

NOTE: If the phonebook is deleted then the terminal will change the **PSC** according to:

Incrementing **PSC** modulus 'FFFFFFFF'.

4.4.2.12.3 EF$_{CC}$ (Change Counter)

The change counter (CC) shall be used to detect changes made to the phone book.

Every update/deletion of an existing phone book entry or the addition of a new phone book entry causes the terminal to increment the EF$_{CC}$. The concept of having a CC makes it possible to update the phone book in different terminals, which still are able to detect the changes (e.g. changes between different handset and/or 2nd and 3rd generation of terminals).

Structure of EF$_{CC}$

Identifier: '4F23'	Structure: transparent	Conditional (see Note)
SFI: '~~XX~~YY'		
File size: 2 bytes	Update activity: high	

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM		

Bytes	Description	M/O	Length
1 to 2	Change Counter (CC) of Phone Book	M	2 bytes
NOTE: This file is mandatory if and only if synchronisation is supported in the phonebook.			

- Change Counter of Phone Book.

    Content:

    - indicates recent change(s) to phone book entries for synchronisation purposes.

    Coding:

    - hexadecimal value. At initialisation, CC shall be personalised to '00 00' (i.e. empty).

4.4.2.12.4 EF$_{PUID}$ (Previous Unique Identifier)

The PUID is used to store the previously used unique identifier (UID). The purpose of this file is to allow the terminal to quickly generate a new UID, which shall then be stored in the EF$_{UID}$.

Structure of EF_{PUID}

Identifier: '4F24'	Structure: transparent		Conditional (see Note)
SFI: '~~XX~~YY'			
File size: 2 bytes		Update activity: high	

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM			

Bytes	Description	M/O	Length
1 to 2	Previous Unique Identifier (PUID) of Phone Book Entry	M	2 bytes
NOTE: This file is mandatory if and only if synchronisation is supported in the phonebook.			

- Previous unique Identifier of Phone Book Entry.

    Content:

    - Previous number that was used to unambiguously identify the phone book entry for synchronisation purposes.

4.4.2.13 EF_{EMAIL} (e-mail address)

This EF contains the e-mail addresses that may be linked to a phone book entry. Several e-mail addresses can be attached to one EF_{ADN} record, using one or several EF_{EMAIL}. The number of email addresses may be equal to or less than the amount of records in EF_{ADN}. Each record contains an e-mail address. The first part indicates the e-mail address, and the second part indicates the reference to the associated record in the EF_{ADN} file. This record cannot be shared between several phonebook entries.

Structure of EF_{EMAIL}

Identifier: '4FXX'	Structure: linear fixed		Optional
SFI: '~~XX~~YY'			
Record length: X or X+2 bytes		Update activity: low	

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM			

Bytes	Description	M/O	Length
1 to X	E-mail Address	M	X bytes
:			
:			
X+1	ADN file SFI	C	1 byte
X+2	ADN file Record Identifier	C	1 byte
NOTE: The fields marked C above are mandatory if and only if the file is not type 1 (as specified in EF_{PBR})			

- E-mail Address.
    Content:

    - string defining the e-mail address

    Coding:

- the SMS default 7-bit coded alphabet as defined in 3G TS 23.038 [5] with bit 8 set to 0. The alpha identifier shall be left justified. Unused bytes shall be set to 'FF'.

- ADN file SFI.

    Content:

    - short File identifier of the associated EF_{ADN} file.

    Coding:

    - as defined in 3G TS 31.101.

- ADN file Record Identifier.

    Content:

    - record identifier of the associated phone book entry.

    Coding:

    - binary.

4.4.2.14 Phonebook restrictions

This subclause lists some general restrictions that apply to the phonebook:

- if an EF_{PBR} file contains more than one record, then they shall all be formatted identically on a type-by-type basis, e.g. if EF_{PBR} record #1 contains one type 1 e-mail then all EF_{PBR} records shall have one type 1 email;

- if an EF_{PBR} record contains more than one reference to one kind of file , such as two EF_{EMAIL} files, then they shall all be formatted identically on a type-by-type basis, e.g. if an EF_{PBR} record has 2 email addresses, then they shall have the same record size and the same number of records in each EF_{PBR} entry;

- an EF_{PBR} record may contain TLV entries indicating that the file exist as a type 1 and 2 file, e.g. a phonebook entry may have two emails, one with a one-to-one mapping (type 1) and one with a indirect mapping (type 2). Regardless of the type, files in all entries shall have the same record configuration.

- an EF_{PBR} record shall not contain more than one occurence of a given kind of file indicated in tag 'AA' (type 3 link). For instance, an EF_{PBR} record may only contain one reference to an EF_{EXT1}.

6.4 User verification and file access conditions

The USIM application uses 2 PINs for user verification, PIN and PIN2. PIN2 is used only in the ADF. The PIN and PIN2 are mapped into key references as defined in 3G TS 31.101 [11]. The Universal PIN shall be associated with a usage qualifier. Other~~Each~~ key references may be~~is~~ associated with a usage qualifier as defined in ISO/IEC7816-9 [26]. The PIN status is indicated in the PS_DO, which is part of the FCP response when an ADF/DF is selected. The coding of the PS_DO is defined in 3G TS 31.101 [11].

PIN and PIN2 are coded on 8 bytes. Only (decimal) digits (0-9) shall be used, coded in CCITT T.50 [23] with bit 8 set to zero. The minimum number of digits is 4. If the number of digits presented by the user is less than 8 then the ME shall pad the presented PIN with 'FF' before sending it to the USIM.

The coding of the UNBLOCK PINs is identical to the coding of the PINs. However, the number of (decimal) digits is always 8.

The security architecture as defined in 3G TS 31.101 [11] applies to the USIM application with the following definitions and additions.

- The USIM application shall use a global key reference ~~'01'~~ as PIN and local key reference ~~'81'~~ as PIN2. For access to DFTelecom the PIN shall be verified. Access with PIN2 is limited to the ADF(USIM) ~~application~~.

- The only valid values for the usage qualifier are~~is~~ '00' (verification requirement is not used) and '08' ~~which means~~ (user authentication knowledge based (PIN)) as defined in ISO/IEC 7816-9 [26]. The terminal shall support the multi-application capabilities as defined in 31.101 [11].

- Every file in the USIM application shall have a reference to an access rule stored in EF_{ARR}.

- Every file under $DF_{Telecom}$ shall have a reference to an access rule stored in EF_{ARR} under $DF_{Telecom}$.

- A multi-application capability UICC (from the security context point of view) shall support the referenced format using SEID as defined in 3G TS 31.101 [11].

- A multi-application capability UICC (from the security context point of view) shall support the replacement of a USIM application PIN with the Universal PIN, key reference '1~~0~~1', as defined in 3G TS 31.101 [11]. Only the Universal PIN is allowed as a replacement.

- A terminal shall support the use of level 1 and level 2 user verification requirements as defined in 3G TS 31.101 [11].

- A terminal shall support the replacement of a USIM application PIN with the Universal PIN, key reference '1~~0~~1', as defined in 3G TS 31.101 [11].

- A terminal shall support the security attributes defined using tag's '8C', 'AB' and '8B' as defined in 3G TS 31.101 [11]. In addition both the referencing methods indicated by tag '8B' shall be supported as defined in 3G TS 31.101 [11].

Disabling of PIN2 is allowed. This is, however, not the case if PIN2 is mapped to the CHV2 of a GSM application.

The access rule is referenced in the FCP using tag '8B'. The TLV object contains the file ID (the file ID of EF_{ARR}) and record number, or file ID (the file ID of EF_{ARR}), SEID and record number, pointer to the record in EF_{ARR} where the access rule is stored. Each SEID refers to a record number in EF_{ARR}. EFs having the same access rule use the same record reference in EF_{ARR}. For a example EF_{ARR}, see 3G TS 31.101 [11].

[…]

7.4 ~~VERIFY command~~

~~The VERIFY command is used to verify the user as defined in 3G TS 31.101 [11]. For the USIM application during a 3G session the parameter P2 is restricted to the following values.~~

~~'01' indicating verification of the PIN;~~

~~'81' indicating verification of PIN2.~~

~~NOTE For administrative purposes any level 5 or level 6 value as specified in 3G TS 31.101 [11] may be used.~~

~~After 3 unsuccessful verification attempts, not necessarily in the same session the PINs blocked. The blocked status is indicated in the response to the VERIFY command (0 attempts left) see 3G TS 31.101 [11].~~

3GPP T3 (USIM) Meeting #20
Marseille, France, 4-5 September, 2001

Tdoc T3-010588

CR-Form-v3

CHANGE REQUEST

⌘   **31.102 CR 099**   ⌘ rev **-** ⌘ Current version: **3.6.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘   (U)SIM **X**   ME/UE **X**   Radio Access Network ☐   Core Network ☐

Title: ⌘	General Corrections	
Source: ⌘	T3	
Work item code: ⌘	TEI	***Date:*** ⌘ 05/09/01
Category: ⌘ **F**		***Release:*** ⌘ R99

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can
be found in 3GPP TR 21.900.

Use one of the following releases:
2   *(GSM Phase 2)*
R96   *(Release 1996)*
R97   *(Release 1997)*
R98   *(Release 1998)*
R99   *(Release 1999)*
REL-4   *(Release 4)*
REL-5   *(Release 5)*

Reason for change: ⌘	
Summary of change: ⌘	Replace 'XX' by 'YY' in the SFI of phonebook files. Section 6.4, replace Universal PIN key reference by '11' Replace "Access with PIN2 is limited to the USIM application" by " Access with PIN2 is limited to the ADF(USIM)" Indicate in section 6.4 that the usage qualifier value '00' is valid. IAP and GRP file update activity set to low Note on the search of the global phonebook
Consequences if not approved: ⌘	Misinterpretation

Clauses affected: ⌘	4.4.2, 6.4

Other specs Affected: ⌘	☐ Other core specifications ⌘	
	☐ Test specifications	
	☐ O&M Specifications	

Other comments: ⌘	

4.4.2 Contents of files at the DF PHONEBOOK level

The UICC may contain a global phonebook, or application specific phonebooks, or both in parallel. When both phonebook types co-exist, they are independent and no data is shared. In this case, it shall be possible for the user to select which phonebook the user would like to access.

It is recommended that the terminal searches for the global phonebook located under $DF_{TELECOM}$ as its presence is not indicated anywhere in the USIM application.

The global phonebook is located in $DF_{PHONEBOOK}$ under $DF_{TELECOM}$. Each specific USIM application phonebook is located in $DF_{PHONEBOOK}$ of its respective Application DF_{USIM}. The organisation of files in $DF_{PHONEBOOK}$ under DF_{USIM} and under $DF_{TELECOM}$ follows the same rules. Yet $DF_{PHONEBOOK}$ under DF_{USIM} may contain a different set of files than $DF_{PHONEBOOK}$ under $DF_{TELECOM}$. All phonebook related EFs are located under their respective $DF_{PHONEBOOK}$. USIM specific phonebooks are dedicated to application specific entries. Each application specific phonebook is protected by the application PIN.

If a GSM application resides on the UICC, the EFs ADN and EXT1 from one $DF_{PHONEBOOK}$ (defined at GSM application installation) are mapped to $DF_{TELECOM}$. Their file IDs are specified in GSM 11.11 [18], i.e. EF_{ADN} = '6F3A' and EF_{EXT1} = '6F4A', respectively. EF_{ADN} and EF_{PBR} shall always be present if the $DF_{Phonebook}$ is present. If any phonebook file other than EF_{ADN} or EF_{EXT1}, is used, then EF_{PBC} shall be present.

If the UICC is inserted into a GSM terminal and a record in the phonebook has been updated, a flag in the entry control information in the EF_{PBC} is set from 0 to 1 by the card. If the UICC is later inserted into a 3G terminal again, the terminal shall check the flag in EF_{PBC} and if this flag is set, shall update the EF_{CC}, and then reset the flag. A set flag in EF_{PBC} results in a full synchronisation of the phonebook between an external entity and the UICC (if synchronisation is requested).

The EF structure related to the public phonebook is located under $DF_{PHONEBOOK}$ in $DF_{TELECOM}$. A USIM specific phonebook may exist for application specific entries. The application specific phonebook is protected by the application PIN. The organisation of files in the application specific phonebook follows the same rules as the one specified for the public phone book under $DF_{TELECOM}$. The application specific phonebook may contain a different set of files than the one in the public area under $DF_{TELECOM}$.

4.4.2.1 EF$_{PBR}$ (Phone Book Reference file)

This file describes the structure of the phonebook. All EFs representing the phonebook are specified here, together with their file identifiers (FID) and their short file identifiers (SFI), if applicable.

Certain kinds of EFs can occur more than once in the phonebook, e.g. there may be two entities of Abbreviated Dialling Numbers, EF_{ADN} and EF_{ADN1}. For these kinds of EFs, no fixed FID values are specified. Instead, the value '4FXX' indicates that the value is to be assigned by the card issuer. These assigned values are then indicated in the associated TLV object in EF_{PBR}.

EFs stating an SFI value ('XXYY') in the description of their structure shall provide an SFI. The value shall be assigned by the card issuer and is indicated in the associated TLV object in EF_{PBR}.

The reference file is a file that contains information how the information in the different files is to be combined together to form a phone book entry. The reference file contains records. Each record specifies the structure of up to 254 entries in the phone book. Each phone book entry consists of data stored in files indicated in the reference file record. The entry structure shall be the same over all the records in the EF $_{PBR}$. If more than 254 entries are to be stored, a second record is needed in the reference file. The structure of a phone book entry is defined by different TLV objects that are stored in a reference file record. The reference file record structure describes the way a record in a file that is part of the phonebook is used to create a complete entry. Three different types of file linking exist.

- Type 1 files: Files that contain as many records as the reference/master file (EF_{ADN}, EF_{ADN1}) and are linked on record number bases (Rec1 -> Rec1). The master file record number is the reference.

- Type 2 files: Files that contain less entries than the master file and are linked via pointers in the index administration file (EF_{IAP}).

- Type 3 files are files that are linked by a record identifier within a record.

Table 4.1: Phone Book Reference file Constructed Tags

Tag Value	Constructed TAG Description
'A8'	Indicating files where the amount of records equal to master EF, type 1
'A9'	Indicating files that are linked using the index administration file, type 2. Order of pointer appearance in index administration EF is the same as the order of file IDs following this tag
'AA'	Indicating files that are linked using a record identifier, type 3. (The file pointed to is defined by the TLV object.)

The first file ID indicated using constructed Tag 'A8' is called the master EF. Access conditions for all other files in the index structure is set to the same as for the master EF unless otherwise specified.

File IDs indicated using constructed Tag 'A8' is a type 1 file and contains the same number of records as the first file that is indicated in the data part of this TLV object. All files following this Tag are mapped one to one using the record numbers/IDs of the first file indicated in this TLV object.

File IDs indicated using constructed Tag 'A9' are mapped to the master EF (the file ID indicated as the first data object in the TLV object using Tag 'A8') using the pointers in the index administration file. The order of the pointers in the index administration file is the same as the order of the file IDs presented after Tag 'A9'. If this Tag is not present in the reference file record the index administration file is not present in the structure. In case the index administration file is not present in the structure it is not indicated in the data following tag 'A8'.

File IDs indicated using constructed Tag 'AA' indicate files that are part of the reference structure but they are addressed using record identifiers within a record in one or more of the files that are part of the reference structure. The length of the tag indicates whether the file to be addressed resides in the same directory or if a path to the file is provided in the TLV object.

Type 2 and type 3 files contain records that may be shared between several phonebook entries (except when otherwise indicated). The terminal shall ensure that a shared record is emptied when the last phonebook entry referencing it is modified in such a way that it doesn't reference the record anymore.

   NOTE: in the current version of the specification, only type 3 files contain records that may be shared.

Each constructed Tag contains a list of primitive Tags indicating the order and the kind of data (e.g. ADN, IAP,…) of the reference structure.

The primitive tag identifies clearly the type of data, its value field indicates the file identifier and, if applicable, the SFI value of the specified EF. That is, the length value of a primitive tag indicates if an SFI value is available for the EF or not:

- Length = '02' Value: 'FID (2 bytes)'

- Length = '03' Value: 'FID (2 bytes)', 'SFI (1 byte)'

Table 4.2: Tag definitions for the phone book kind of file

Tag Value	TAG Description
'C0'	EF$_{ADN}$ data object
'C1'	EF$_{IAP}$ data object
'C2'	EF$_{EXT1}$ data object
'C3'	EF$_{SNE}$ data object
'C4'	EF$_{ANR}$ data object
'C5'	EF$_{PBC}$ data object
'C6'	EF$_{GRP}$ data object
'C7'	EF$_{AAS}$ data object
'C8'	EF$_{GAS}$ data object
'C9'	EF$_{UID}$ data object
'CA'	EF$_{EMAIL}$ data object
'CB'	EF$_{CCP1}$ data object

Table 4.3 (below) lists the allowed types for each kind of file:

Table 4.3: Presence of files as type

File name	Type 1	Type 2	Type 3
EF$_{AAS}$			X
EF$_{ADN}$	X		
EF$_{ANR}$	X	X	
EF$_{EMAIL}$	X	X	
EF$_{EXT1}$			X
EF$_{GAS}$			X
EF$_{GRP}$	X		
EF$_{IAP}$	X		
EF$_{PBC}$	X		
EF$_{SNE}$	X	X	
EF$_{UID}$	X		
EF$_{CCP1}$			X

Phone Book Reference file EF$_{PBR}$ structure

Identifier: '4F30'		Structure: linear fixed		Conditional (see Note)
Record Length: X bytes		Update activity: low		
Access Conditions: 　　READ　　　　　　　PIN 　　UPDATE　　　　　ADM 　　DEACTIVATE　　ADM 　　ACTIVATE　　　　ADM				
Bytes	Description		M/O	Length
1 to X	TLV object(s) for indicating EFs that are part of the phone book structure		M	X bytes
NOTE: This file is mandatory if and only if DF$_{Phonebook}$ is present.				

At the end of each record, unused bytes, if any, shall be filled with 'FF'.

4.4.2.2　　　EF$_{IAP}$ (Index Administration Phone book)

This file is present if Tag 'A9' is indicated in the reference file.

The EF contains pointers to the different records in the files that are part of the phone book. The index administration file record number/ID is mapped one to one with the corresponding EF$_{ADN}$ (shall be record to record). The index administration file contains the same amount of records as EF$_{ADN}$. The order of the pointers in an EF$_{IAP}$ shall be the same as the order of file IDs that appear in the TLV object indicated by Tag 'A9' in the reference file record. The amount of bytes in a record is equal to the number of files indicated the EF$_{PBR}$ following tag 'A9'.

The value 'FF' is an invalid record number/ID and is used in any location in to indicate that no corresponding record in the indicated file is available.

The content of EF$_{IAP}$ is set to 'FF' at the personalisation stage.

Index administration file EF$_{IAP}$ structure

Identifier: '4FXX'	Structure: linear fixed	Conditional (see Note)
SFI: '~~XX~~YY'		
Record Length: X bytes	Update activity: low~~high~~	

Access Conditions:	
READ	PIN
UPDATE	PIN
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1	Record number of the first object indicated after Tag 'A9'	M	1 byte
2	Record number of the second object indicated after Tag 'A9'	M	1 byte
X	Record number of the xth object indicated after Tag 'A9'	M	1 byte
NOTE:	This file is mandatory if and only if type 2 files are present.		

4.4.2.3   EF$_{ADN}$ (Abbreviated dialling numbers)

This EF contains Abbreviated Dialling Numbers (ADN) and/or Supplementary Service Control strings (SSC). In addition it contains identifiers of associated network/bearer capabilities and identifiers of extension records. It may also contain an associated alpha-tagging.

Identifier: '4FXX'	Structure: linear fixed	Conditional (see Note)
SFI: '~~XX~~YY'		
Record length: X+14 bytes	Update activity: low	

Access Conditions:	
READ	PIN
UPDATE	PIN
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1 to X	Alpha Identifier	O	X bytes
X+1	Length of BCD number/SSC contents	M	1 byte
X+2	TON and NPI	M	1 byte
X+3 to X+12	Dialling Number/SSC String	M	10 bytes
X+13	Capability/Configuration1 Identifier	M	1 byte
X+14	Extension1 Record Identifier	M	1 byte
NOTE:	This file is mandatory if and only if DF$_{PHONEBOOK}$ is present.		

- Alpha Identifier.
  Contents:
  - Alpha-tagging of the associated dialling number.
  Coding:
  - this alpha-tagging shall use
    either:
    - the SMS default 7-bit coded alphabet as defined in 3G TS 23.038 [5] with bit 8 set to 0. The alpha identifier shall be left justified. Unused bytes shall be set to 'FF'.
    or:
    - one of the UCS2 coded options as defined in the annex of 3G TS 31.101 [11].

NOTE 1: The value of X may be from zero to 241. Using the command GET RESPONSE the ME can determine the value of X.

- Length of BCD number/SSC contents.
  Contents:
  - this byte gives the number of bytes of the following two data items containing actual BCD number/SSC information. This means that the maximum value is 11, even when the actual ADN/SSC information length is greater than 11. When an ADN/SSC has extension, it is indicated by the extension1 identifier being unequal to 'FF'. The remainder is stored in the EF_{EXT1} with the remaining length of the additional data being coded in the appropriate additional record itself (see subclause 4.4.2.4).
  Coding:
  - according to 3G TS 24.008 [9].

- TON and NPI.
  Contents:
  - Type of number (TON) and numbering plan identification (NPI).
  Coding:
  - according to 3G TS 24.008 [9]. If the Dialling Number/SSC String does not contain a dialling number, e.g. a control string deactivating a service, the TON/NPI byte shall be set to 'FF' by the ME (see note 2).

NOTE 2: If a dialling number is absent, no TON/NPI byte is transmitted over the radio interface (see 3G TS 24.008 [9]). Accordingly, the ME should not interpret the value 'FF' and not send it over the radio interface.
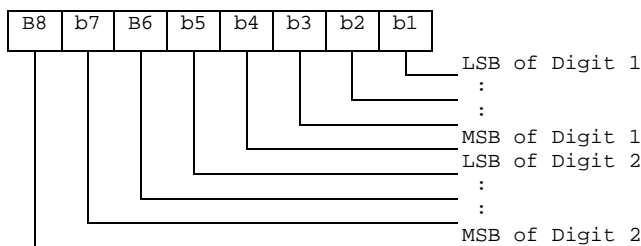

- Dialling Number/SSC String
  Contents:
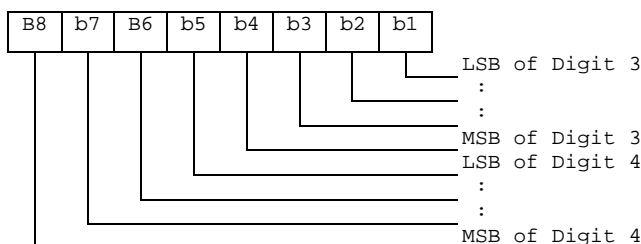 - up to 20 digits of the telephone number and/or SSC information.
  Coding:
  - according to 3G TS 24.008 [9] , 3G TS 22.030 [4] and the extended BCD-coding (see table 4.4). If the telephone number or SSC is longer than 20 digits, the first 20 digits are stored in this data item and the remainder is stored in an associated record in the EF_{EXT1}. The record is identified by the Extension1 Record Identifier. If ADN/SSC require less than 20 digits, excess nibbles at the end of the data item shall be set to 'F'. Where individual dialled numbers, in one or more records, of less than 20 digits share a common appended digit string the first digits are stored in this data item and the common digits stored in an associated record in the EF_{EXT1}. The record is identified by the Extension 1 Record Identifier. Excess nibbles at the end of the data item shall be set to 'F'.

Byte X+3

Byte X+4:

etc.

- Capability/Configuration1 Identifier.
  Contents:
  - capability/configuration identification byte. This byte identifies the number of a record in the EF$_{CCP1}$ containing associated capability/configuration parameters required for the call. The use of this byte is optional. If it is not used it shall be set to 'FF'.
  Coding:
  - binary.

- Extension1 Record Identifier.
  Contents:
  - extension1 record identification byte. This byte identifies the number of a record in the EF$_{EXT1}$ containing an associated called party subaddress or additional data. The use of this byte is optional. If it is not used it shall be set to 'FF'.
  - if the ADN/SSC requires both additional data and called party subaddress, this byte identifies the additional record. A chaining mechanism inside EF$_{EXT1}$ identifies the record of the appropriate called party subaddress (see subclause 4.4.2.4).
  Coding:
  - binary.

NOTE 3: EF$_{ADN}$ in the public phone book under DF$_{TELECOM}$ may be used by USIM, GSM and also other applications in a multi-application card. If the non-GSM application does not recognise the use of Type of Number (TON) and Number Plan Identification (NPI), then the information relating to the national dialling plan shall be held within the data item dialling number/SSC and the TON and NPI fields set to UNKNOWN. This format would be acceptable for 3G operation and also for the non-GSM application where the TON and NPI fields shall be ignored.

EXAMPLE: SIM storage of an International Number using E.164 [22] numbering plan.

	TON	NPI	Digit field.
USIM application	001	0001	abc...
Other application compatible with 3G	000	0000	xxx...abc...

where "abc..." denotes the subscriber number digits (including its country code), and "xxx..." denotes escape digits or a national prefix replacing TON and NPI.

NOTE 4: When the ME acts upon the EF$_{ADN}$ with a SEARCH RECORD command in order to identify a character string in the alpha-identifier, it is the responsibility of the ME to ensure that the number of characters used as SEARCH RECORD parameters are less than or equal to the value of X if the MMI allows the user to offer a greater number.

Table 4.4: Extended BCD coding

BCD Value	Character/Meaning
'0'	"0"
:	:
'9'	"9"
'A'	"*"
'B'	"#"
'C'	DTMF Control digit separator (GSM 02.07 [17]).
'D'	"Wild" value. This will cause the MMI to prompt the user for a single digit (see GSM 02.07 [17]).
'E'	RFU.
'F'	Endmark e.g. in case of an odd number of digits.

BCD values 'C', 'D' and 'E' are never sent across the radio interface.

NOTE 5: A second or subsequent 'C' BCD value will be interpreted as a 3 second PAUSE (see GSM 02.07 [17]).
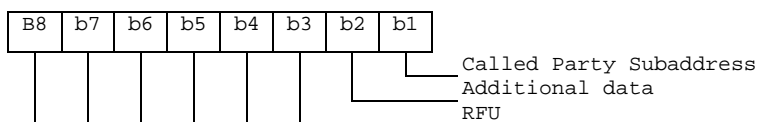
4.4.2.4 EF$_{EXT1}$ (Extension1)

This EF contains extension data of an ADN/SSC. . This EF shall always be present if the DF$_{Phonebook}$ is present.

Extension data is caused by:

- an ADN/SSC which is greater than the 20 digit capacity of the ADN/SSC Elementary File or where common digits are required to follow an ADN/SSC string of less than 20 digits. The remainder is stored in this EF as a record, which is identified by a specified identification byte inside the ADN/SSC Elementary File. The EXT1 record in this case is specified as additional data;

- an associated called party subaddress. The EXT1 record in this case is specified as subaddress data.
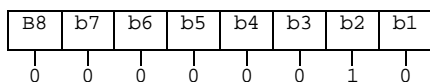
Identifier: '4FXX'		Structure: linear fixed	Optional
SFI: 'XXYY'			
Record length: 13 bytes		Update activity: low	

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM			

Bytes	Description	M/O	Length
1	Record type	M	1 byte
2 to 12	Extension data	M	11 bytes
13	Identifier	M	1 byte

- Record type.
  Contents:
  - type of the record.
  Coding:

B8	b7	b6	b5	b4	b3	b2	b1

Called Party Subaddress
Additional data
RFU

- b3-b8 are reserved and set to 0;

- a bit set to 1 identifies the type of record;

- only one type can be set;

- '00' indicates the type "unknown".

The following example of coding means that the type of extension data is "additional data":

B8	b7	b6	b5	b4	b3	b2	b1
0	0	0	0	0	0	1	0

- Extension data.
  Contents:
    additional data or Called Party Subaddress depending on record type.
  Coding:
    Case 1, Extension1 record is additional data:
    - The first byte of the extension data gives the number of bytes of the remainder of ADN/SSC. The coding of remaining bytes is BCD, according to the coding of ADN/SSC. Unused nibbles at the end shall be set to 'F'. It is possible if the number of additional digits exceeds the capacity of the additional record to chain another record inside the EXT1 Elementary File by the identifier in byte 13.

Case 2, Extension1 record is Called Party Subaddress:
- The subaddress data contains information as defined for this purpose in 3G TS 24.008 [9]. All information defined in 3G TS 24.008, except the information element identifier, shall be stored in the USIM. The length of this subaddress data can be up to 22 bytes. In those cases where two extension records are needed, these records are chained by the identifier field. The extension record containing the first part of the called party subaddress points to the record which contains the second part of the subaddress.

- Identifier.
  Contents:
  identifier of the next extension record to enable storage of information longer than 11 bytes.
  Coding:
  record number of next record. 'FF' identifies the end of the chain.

- Example of a chain of extension records being associated to an ADN/SSC. The extension1 record identifier (Byte 14+X) of ADN/SSC is set to 3.

No of Record	Type	Extension Data	Next	Record
:	:	:	:	
:	:	:	:	
Record 3	'02'	xx xx	'06'	▶
Record 4	'xx'	xx xx	'xx'	
Record 5	'01'	xx xx	'FF'	◀
Record 6	'01'	xx xx	'05'	◀
:	:	:	:	
:	:	:	:	

In this example ADN/SSC is associated to additional data (record 3) and a called party subaddress whose length is more than 11 bytes (records 6 and 5).

## 4.4.2.5	EF~PBC~ (Phone Book Control)

This EF contains control information related to each entry in the phone book. This EF contains as many records as the EF~ADN~ associated with it (shall be record to record). Each record in EF~PBC~ points to a record in its EF~ADN~. This file indicates the control information and the hidden information of each phone book entry.
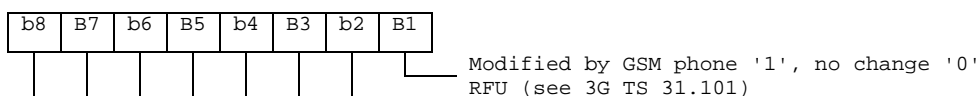
The content of EF~PBC~ is linked to the associated EF~ADN~ record by means of the ADN record number/ID (there is a one to one mapping of record number/identifiers between EF~PBC~ and EF~ADN~).

Structure of control file EF~PBC~

Identifier: '4FXX'	Structure: linear fixed		Conditional (see Note)
SFI: '~~XX~~YY'			
Record length: 2 bytes		Update activity: low	
Access Conditions: 　READ　　　　　PIN 　UPDATE　　　　PIN 　DEACTIVATE　　ADM 　ACTIVATE　　　ADM			
Bytes	Description	M/O	Length
1	Entry Control Information	M	1 byte
2	Hidden Information	M	1 byte
NOTE:　　This file is mandatory if and only if one or both of the following is true: 　　　　　- hidden entries are supported 　　　　　- a GSM SIM application is supported in the UICC.			

- Entry Control Information.
  Contents:
  - provides some characteristics about the phone book entry (eg modification by a GSM mobile).

Coding:

b8	B7	b6	B5	b4	B3	b2	B1

Modified by GSM phone '1', no change '0'
RFU (see 3G TS 31.101)

- Hidden Information.

    Contents:

    indicates to which USIM application of the UICC this phone book entry belongs, so that the corresponding secret code can be verified to display the phone book entry. If the secret code is not verified, then the phone book entry is hidden.

    Coding:

    '00' – the phone book entry is not hidden;

    'xx' – the phone book entry is hidden. 'xx' is the record number in EF_{DIR} of the associated USIM application.

4.4.2.6 EF$_{GRP}$ (Grouping file)

This EF contains the grouping information for each phone book entry. This file contains as many records as the associated EF_{ADN}. Each record contains a list of group identifiers, where each identifier can reference a group to which the entry belongs.

Structure of grouping file EF$_{GRP}$

Identifier: '4FXX'		Structure: linear fixed		Conditional (see Note)
SFI: '~~XX~~YY'				
Record Length: X bytes (1 ≤ X ≤10)			Update activity: low~~high~~	
Access Conditions: READ PIN UPDATE PIN DEACTIVATE ADM ACTIVATE ADM				
Bytes	Description		M/O	Length
1	Group Name Identifier 1		M	1 byte
2	Group Name Identifier 2		O	1 byte
X	Group Name Identifier X		O	1 byte
NOTE: This file is mandatory if and only if EF$_{GAS}$ is present.				

- Group Name Identifier x.

    Content:

    - indicates if the associated entry is part of a group, in that case it contains the record number of the group name in EF_{GAS}.

    - One entry can be assigned to a maximum of 10 groups.

    Coding:

    - '00' – no group indicated;

        'XX' – record number in EF_{GAS} containing the alpha string naming the group of which the phone book entry is a member.

4.4.2.7 EF$_{AAS}$ (Additional number Alpha String)

This file contains the alpha strings that are associated with the user defined naming tags for additional numbers referenced in EF_{ANR}.

Structure of EF_{AAS}

Identifier: '4FXX'		Structure: linear fixed		Optional
SFI: -				
Record length: X bytes		Update activity: low		

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM				

Bytes	Description		M/O	Length
1 to X	Alpha text string		M	X bytes

- Alpha text string.

    Content:

    - user defined text for additional number.

    Coding:

    - same as the alpha identifier in EF_{ADN}.

4.4.2.8   EF_{GAS} (Grouping information Alpha String)

This file contains the alpha strings that are associated with the group name referenced in EF_{GRP}.

Structure of EF_{GAS}

Identifier: '4FXX'		Structure: linear fixed		Conditional (see Note)
SFI: -				
Record length: X bytes		Update activity: low		

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM				

Bytes	Description		M/O	Length
1 to X	Alpha text string		M	X bytes
NOTE: This file is mandatory if and only if EF_{GRP} is present.				

- Alpha text string

    Content:

    - group names.

    Coding:

    - same as the alpha identifier in EF_{ADN}.

4.4.2.9   EF_{ANR} (Additional Number)

Several phone numbers and/or Supplementary Service Control strings (SSC) can be attached to one EF_{ADN} record, using one or several EF_{ANR}. The amount of additional number entries may be less than or equal to the amount of records in EF_{ADN}. The EF structure is linear fixed. Each record contains an additional phone number or Supplementary Service Control strings (SSC). This record cannot be shared between several phonebook entries. The first byte indicates whether

the record is free or the type of additional number referring to the record number in EF$_{AAS}$, containing the text to be displayed. The following part indicates the additional number and the reference to the associated record in the EF$_{ADN}$ file. In addition it contains identifiers of associated network/bearer capabilities and identifiers of extension records.

Structure of EF$_{ANR}$

Identifier: '4FXX'		Structure: linear fixed		Optional
SFI: '~~XX~~YY'				
Record length: 15 or 17 bytes			Update activity: low	

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM				

Bytes	Description		M/O	Length
1	Additional Number identifier		M	1 byte
2	Length of BCD number/SSC contents		M	1 byte
3	TON and NPI		M	1 byte
4 to 13	Additional number/SSC String		M	10 bytes
14	Capability/Configuration1 Identifier		M	1 byte
15	Extension1 Record Identifier		M	1 byte
16	ADN file SFI		C	1 byte
17	ADN file Record Identifier		C	1 byte
NOTE: The fields marked C above are mandatory if and only if the file is not type 1 (as specified in EF$_{PBR}$)				

- Additional Number Identifier

    Content:

    - describes the type of the additional number defined in the file EF$_{AAS}$.

    Coding:

    - '00' – no additional number description;
      'xx' – record number in EF$_{AAS}$ describing the type of number (e.g. "FAX");
      'FF' – free record.

- Length of BCD number/SSC contents

    Contents:

    - this byte gives the number of bytes of the following two data items containing actual BCD number/SSC information. This means that the maximum value is 11, even when the actual additional number/SSC information length is greater than 11. When the additional number/SSC has extension, it is indicated by the extension1 identifier being unequal to 'FF'. The remainder is stored in the EF$_{EXT1}$ with the remaining length of the additional data being coded in the appropriate additional record itself (see subclause 4.4.2.4).

    Coding:

    - same as the length of BCD number/SSC string byte in EF$_{ADN.}$

- TON and NPI.

    Contents:

    - Type of number (TON) and numbering plan identification (NPI).

    Coding:

    - same as the TON and NPI byte in EF$_{ADN.}$

- Additional number/SSC string

    Content:

    - up to 20 digits of the additional phone number and/or SSC information linked to the phone book entry.

    Coding:

    - same as the dialling number /SSC string in EF$_{ADN.}$

- Capability/Configuration1 Identifier.

    Contents:

    - This byte identifies the number of a record in the EF$_{CCP1}$ containing associated capability/configuration parameters required for the call. The use of this byte is optional. If it is not used it shall be set to 'FF'.

    Coding:

    - binary.

- Extension1 Record Identifier.

    Contents:

    - extension1 record identification byte. This byte identifies the number of a record in the EF$_{EXT1}$ containing an associated called party subaddress or additional data. The use of this byte is optional. If it is not used it shall be set to 'FF'.

        if the number requires both additional data and called party subaddress, this byte identifies the additional record. A chaining mechanism inside EF$_{EXT1}$ identifies the record of the appropriate called party subaddress (see subclause 4.4.2.4).

    Coding:

    - binary.

- ADN file SFI.

    Content:

    - Short File identifier of the associated EF$_{ADN}$ file.

    Coding:

    - as defined in the UICC specification.

- ADN file Record Identifier

    Content:

    - record identifier of the associated phone book entry.

    Coding:

    - 'xx' – record identifier of the corresponding ADN record.

4.4.2.10 EF$_{SNE}$ (Second Name Entry)

The phone book also contains the option of a second name entry. The amount of second name entries may be less than or equal to the amount of records in EF$_{ADN}$. Each record contains a second name entry. This record cannot be shared between several phonebook entries.

Structure of EF$_{SNE}$

Identifier: '4FXX'		Structure: linear fixed		Optional
SFI: '~~XX~~YY'				
Record length: X or X+2 bytes		Update activity: low		

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM				

Bytes	Description		M/O	Length
1 to X	Alpha Identifier of Second Name		M	X bytes
X+1	ADN file SFI		C	1 byte
X+2	ADN file Record Identifier		C	1 byte
NOTE: The fields marked C above are mandatory if and only if the file is not type 1 (as specified in EF$_{PBR}$)				

- Alpha Identifier of Second Name.

    Content:

    - string defining the second name of the phone book entry.

    Coding:

    - as the alpha identifier for EF$_{ADN}$.

- ADN file SFI.

    Content:

    - Short File identifier of the associated EF$_{ADN}$ file.

    Coding:

    - as defined in the UICC specification.

- ADN file Record Identifier

    Content:

    record identifier of the associated phone book entry.

    Coding:

    'xx' – record identifier of the corresponding ADN record.

4.4.2.11   EF$_{CCP1}$ (Capability Configuration Parameters 1)

This EF contains parameters of required network and bearer capabilities and ME configurations associated with a call established using a phone book entry.

Structure of EF_{CCP1}

Identifier: '''4FXX'	Structure: linear fixed		Optional
SFI: '~~XX~~YY'			
Record length: X bytes, X ≥ 15	Update activity: low		

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM			

Bytes	Description	M/O	Length
1 to X	Bearer capability information element	M	X bytes

- Bearer capability information element.

    Contents and Coding:

    - see 3G TS 24.008 [9]. The Information Element Identity (IEI) shall be excluded; i.e. the first byte of the EF_{CCP1} record shall be Length of the bearer capability contents.

"- unused bytes are filled with 'FF'

4.4.2.12 Phone Book Synchronisation

To support synchronisation of phone book data with other devices, the USIM may provide the following files to be used by the synchronisation method: a phone book synchronisation counter (PSC), a unique identifier (UID) and change counter (CC) to indicate recent changes.

If synchronisation is supported in the phonebook, then EF_{PSC}, EF_{UID}, EF_{PUID} and EF_{CC} are all mandatory.

4.4.2.12.1 EF_{UID} (Unique Identifier)

The EF_{UID} is used to uniquely identify a record and to be able to keep track of the entry in the phone book. The terminal assigns the (UID) when a new entry is created. The value of the UID does not change as long as the value of the PBID remains the same. The UID shall remain on the UICC, in EF_{UID}, until the PBID is regenerated. This means that when a phone book entry is deleted, the content of the linked information (eg ADN, E-MAIL,..) shall be set to the personalization value 'FF…FF'. But the UID-value of the deleted record shall not be used when a new entry is added to the phonebook until the PBID is regenerated, but it shall be set to a new value.

If/when the PBID is regenerated, all UIDs for the entry in the phone book shall be assigned new values starting from 1. The new value of the UID for each entry shall then be kept until the PBID is regenerated again.

Structure of EF_{UID}

Identifier: '4FXX'	Structure: linear fixed		Conditional (see Note)
SFI: '~~XX~~YY'			
Record length: 2 bytes	Update activity: low		

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM			

Bytes	Description	M/O	Length
1 to 2	Unique Identifier (UID) of Phone Book Entry	M	2 bytes
NOTE: This file is mandatory if and only if synchronisation is supported in the phonebook.			

- Unique Identifier of Phone Book Entry.

    Content:

    - number to unambiguously identify the phone book entry for synchronisation purposes.

    Coding:

    - hexadecimal value. At initialisation all UIDs are personalised to "00 00" (i.e. empty).

4.4.2.12.2 EF$_{PSC}$ (Phone book Synchronisation Counter)

The phone book synchronisation counter (PSC) is used by the ME to construct the phone book identifier (PBID) and to determine whether the accessed phone book is the same as the previously accessed phone book or if it is a new unknown phone book (might be the case that there is one phonebook under DF-telecom and one phone book residing in a USIM-application). If the PSC is unknown, a full synchronisation of the phone book will follow.

The PSC is also used to regenerate the UIDs and reset the CC to prevent them from running out of range. When the UIDs or the CC has reached its maximum value, a new PSC is generated. This leads to a scenario where neither the CC nor the UIDs will run out of range.

The PSC shall be regenerated by the terminal if one of the following situation applies:
- the values of the UIDs have run out of range;
- the whole phone book has been reset/deleted;
- the value of the CC has run out of range.

Structure of EF$_{PSC}$

Identifier: '4F22'	Structure: transparent		Conditional (see Note)
SFI: '~~XX~~YY'			
File size: 4 bytes		Update activity: low	

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM			

Bytes	Description	M/O	Length
1 to 4	Phone book synchronisation counter (PSC)	M	4 bytes
NOTE: This file is mandatory if and only if synchronisation is supported in the phonebook.			

- PSC: Unique synchronisation counter of Phone Book.

    Content:

    number to unambiguously identify the status of the phone book for synchronisation purposes.

    Coding:

    hexadecimal value.

    The phone book identifier (PBID) coding based on the EF$_{PSC}$ is described hereafter:

- For a phone book residing in DF-telecom:

    - PBID = ICCid (10bytes) "fixed part" + 4 bytes (in EF$_{PSC}$) "variable part".

- For a phone book residing in an USIM application:

    - PBID = 10 last bytes of (ICCid XOR AID) "fixed part" + 4 bytes (in EF$_{PSC}$) "variable part".

To be able to detect if the PSC needs to be regenerated (i.e. the variable part) the following test shall be made by the terminal before for each update of either the CC or the assignment of a new UID:

- Each time the terminal has to increment the value of the UID the following test is needed:

    - If UID = 'FF FF' then.

        {Increment **PSC** mod 'FF FF FF FF'; all the UIDs shall be regenerated}.

- Each time the terminal has to increment the value of CC the following test is needed:

    If CC = 'FF FF' then.

    {Increment **PSC** mod 'FF FF FF FF' ; CC=0001}.

NOTE: If the phonebook is deleted then the terminal will change the **PSC** according to:

Incrementing **PSC** modulus 'FFFFFFFF'.

4.4.2.12.3   EF$_{CC}$ (Change Counter)

The change counter (CC) shall be used to detect changes made to the phone book.

Every update/deletion of an existing phone book entry or the addition of a new phone book entry causes the terminal to increment the EF$_{CC}$. The concept of having a CC makes it possible to update the phone book in different terminals, which still are able to detect the changes (e.g. changes between different handset and/or 2nd and 3rd generation of terminals).

Structure of EF$_{CC}$

Identifier: '4F23'	Structure: transparent		Conditional (see Note)
SFI: '~~XX~~YY'			
File size: 2 bytes		Update activity: high	

Access Conditions:
    READ   PIN
    UPDATE   PIN
    DEACTIVATE   ADM
    ACTIVATE ADM			

Bytes	Description	M/O	Length
1 to 2	Change Counter (CC) of Phone Book	M	2 bytes
NOTE: This file is mandatory if and only if synchronisation is supported in the phonebook.			

- Change Counter of Phone Book.

    Content:

    - indicates recent change(s) to phone book entries for synchronisation purposes.

    Coding:

    - hexadecimal value. At initialisation, CC shall be personalised to '00 00' (i.e. empty).

4.4.2.12.4   EF$_{PUID}$ (Previous Unique Identifier)

The PUID is used to store the previously used unique identifier (UID). The purpose of this file is to allow the terminal to quickly generate a new UID, which shall then be stored in the EF$_{UID}$.

Structure of EF$_{PUID}$

Identifier: '4F24'	Structure: transparent		Conditional (see Note)
SFI: '~~XX~~YY'			
File size: 2 bytes		Update activity: high	

Access Conditions:
     READ   PIN
     UPDATE   PIN
     DEACTIVATE   ADM
     ACTIVATE ADM			

Bytes	Description	M/O	Length
1 to 2	Previous Unique Identifier (PUID) of Phone Book Entry	M	2 bytes
NOTE: This file is mandatory if and only if synchronisation is supported in the phonebook.			

- Previous unique Identifier of Phone Book Entry.

  Content:

  - Previous number that was used to unambiguously identify the phone book entry for synchronisation purposes.

4.4.2.13 EF$_{EMAIL}$ (e-mail address)

This EF contains the e-mail addresses that may be linked to a phone book entry. Several e-mail addresses can be attached to one EF$_{ADN}$ record, using one or several EF$_{EMAIL}$. The number of email addresses may be equal to or less than the amount of records in EF$_{ADN}$. Each record contains an e-mail address. The first part indicates the e-mail address, and the second part indicates the reference to the associated record in the EF$_{ADN}$ file. This record cannot be shared between several phonebook entries.

Structure of EF$_{EMAIL}$

Identifier: '4FXX'	Structure: linear fixed		Optional
SFI: '~~XX~~YY'			
Record length: X or X+2 bytes		Update activity: low	

Access Conditions:
     READ   PIN
     UPDATE   PIN
     DEACTIVATE   ADM
     ACTIVATE ADM			

Bytes	Description	M/O	Length
1 to X	E-mail Address	M	X bytes
:			
:			
X+1	ADN file SFI	C	1 byte
X+2	ADN file Record Identifier	C	1 byte
NOTE: The fields marked C above are mandatory if and only if the file is not type 1 (as specified in EF$_{PBR}$)			

- E-mail Address.
  Content:

  - string defining the e-mail address

  Coding:

- the SMS default 7-bit coded alphabet as defined in 3G TS 23.038 [5] with bit 8 set to 0. The alpha identifier shall be left justified. Unused bytes shall be set to 'FF'.

- ADN file SFI.

    Content:

    - short File identifier of the associated EF_{ADN} file.

    Coding:

    - as defined in 3G TS 31.101.

- ADN file Record Identifier.

    Content:

    - record identifier of the associated phone book entry.

    Coding:

    - binary.

4.4.2.14 Phonebook restrictions

This subclause lists some general restrictions that apply to the phonebook:

- if an EF_{PBR} file contains more than one record, then they shall all be formatted identically on a type-by-type basis, e.g. if EF_{PBR} record #1 contains one type 1 e-mail then all EF_{PBR} records shall have one type 1 email;

- if an EF_{PBR} record contains more than one reference to one kind of file , such as two EF_{EMAIL} files, then they shall all be formatted identically on a type-by-type basis, e.g. if an EF_{PBR} record has 2 email addresses, then they shall have the same record size and the same number of records in each EF_{PBR} entry;

- an EF_{PBR} record may contain TLV entries indicating that the file exist as a type 1 and 2 file, e.g. a phonebook entry may have two emails, one with a one-to-one mapping (type 1) and one with a indirect mapping (type 2). Regardless of the type, files in all entries shall have the same record configuration.

- an EF_{PBR} record shall not contain more than one occurence of a given kind of file indicated in tag 'AA' (type 3 link). For instance, an EF_{PBR} record may only contain one reference to an EF_{EXT1}.

6.4 User verification and file access conditions

The USIM application uses 2 PINs for user verification, PIN and PIN2. PIN2 is used only in the ADF. The PIN and PIN2 are mapped into key references as defined in 3G TS 31.101 [11]. The Universal PIN shall be associated with a usage qualifier. OtherEach key references may beis associated with a usage qualifier as defined in ISO/IEC7816-9 [26]. The PIN status is indicated in the PS_DO, which is part of the FCP response when an ADF/DF is selected. The coding of the PS_DO is defined in 3G TS 31.101 [11].

PIN and PIN2 are coded on 8 bytes. Only (decimal) digits (0-9) shall be used, coded in CCITT T.50 [23] with bit 8 set to zero. The minimum number of digits is 4. If the number of digits presented by the user is less than 8 then the ME shall pad the presented PIN with 'FF' before sending it to the USIM.

The coding of the UNBLOCK PINs is identical to the coding of the PINs. However, the number of (decimal) digits is always 8.

The security architecture as defined in 3G TS 31.101 [11] applies to the USIM application with the following definitions and additions.

- The USIM application shall use key reference '01' as PIN and key reference '81' as PIN2. For access to DFTelecom the PIN shall be verified. Access with PIN2 is limited to the ADF(USIM) application.

- The only valid values for the usage qualifier are~~is~~ '00' (verification requirement is not used) and '08' ~~which means~~ (user authentication knowledge based (PIN)) as defined in ISO/IEC 7816-9 [26]. The terminal shall support the multi-application capabilities as defined in 31.101 [11].

- Every file in the USIM application shall have a reference to an access rule stored in EF$_{ARR}$.

- Every file under DF$_{Telecom}$ shall have a reference to an access rule stored in EF$_{ARR}$ under DF$_{Telecom}$.

- A multi-application capability UICC (from the security context point of view) shall support the referenced format using SEID as defined in 3G TS 31.101 [11].

- A multi-application capability UICC (from the security context point of view) shall support the replacement of a USIM application PIN with the Universal PIN, key reference '1~~0~~1', as defined in 3G TS 31.101 [11]. Only the Universal PIN is allowed as a replacement.

- A terminal shall support the use of level 1 and level 2 user verification requirements as defined in 3G TS 31.101 [11].

- A terminal shall support the replacement of a USIM application PIN with the Universal PIN, key reference '1~~0~~1', as defined in 3G TS 31.101 [11].

- A terminal shall support the security attributes defined using tag's '8C', 'AB' and '8B' as defined in 3G TS 31.101 [11]. In addition both the referencing methods indicated by tag '8B' shall be supported as defined in 3G TS 31.101 [11].

Disabling of PIN2 is allowed. This is, however, not the case if PIN2 is mapped to the CHV2 of a GSM application.

The access rule is referenced in the FCP using tag '8B'. The TLV object contains the file ID (the file ID of EF$_{ARR}$) and record number, or file ID (the file ID of EF$_{ARR}$), SEID and record number, pointer to the record in EF$_{ARR}$ where the access rule is stored. Each SEID refers to a record number in EF$_{ARR}$. EFs having the same access rule use the same record reference in EF$_{ARR}$. For a example EF$_{ARR}$, see 3G TS 31.101 [11].